

SENSIO LABS

Mastering Symphony & Doctrine Training

Sensio

92-98 boulevard Victor Hugo

92115 Clichy

SOMMAIRE

Introduction	3
The Symfony framework	3
Training goals	3
Public and technical prerequisites	4
Duration and prices	4
Pedagogic means	4
The trainer	5
How does the training take place?	5
Detailed program	6
From a flat application to an MVC based application	6
Templating with Symfony	6
Symfony development environment	7
The routing system	7
Object Relational Mapping Abstraction	8
Framework Installation	8
Admin Generator	9
Authentication and ACL	9
Internationalization and localization	10
Automated tests	10

INTRODUCTION

The « **Mastering Symfony and Doctrine** » training allows you to learn all framework fundamentals in three days. At the end of the training, you will be able to work on Symfony 1.4 project on your own.

During three days, you will be trained by a Sensio Labs consultant, who's a PHP 5 and Symfony expert. He will help to start working with Symfony thanks to lots of practical exercises covering several framework features. You will discover how the MVC pattern is integrated to Symfony, how to validate your code with bunches of unit and functional tests or how to request a database thanks to the Doctrine library.

THE SYMFONY FRAMEWORK

Symfony is a full stack PHP 5 based on Sensio Labs experience with lots of big companies project. Symfony was created to answer to professional website needs and to bring professional and Open-Source worlds closer.

Nowadays, Symfony is a mature framework, which allows to build and industrialize web developments. Its main advantages are listed below:

- Insuring code structure and standardization ,
- Industrializing developments,
- Insuring maintenance operations and long life to PHP projects.

Sensio Labs is also at the origin of the official documentation and published several books like the « The Definitive Guide to Symfony » by APress publishing. In 2009, this documentation was updated for versions 1.2, 1.3 and 1.4 with the « Practical Symfony » book.

TRAINING GOALS

The main goal of this training consists to transmit all symfony fundamental knowledge to attendees in order to be able to work on their own with the framework. To do this, the training follows a pedagogic way with around 70% of the time dedicated to practical exercises.

PUBLIC AND TECHNICAL PREREQUISITES

This training is destined to web developers and project managers having a good experience with the PHP programming language, object oriented programming and databases. These prerequisites are mandatory if you want to take benefit from the training during the three days.

DURATION AND PRICES

The training takes place during three days (21 hours) in a meeting room.

The price per attendee is **1 400 euros (without taxes)**.

The price per attendee is **1 200 euros (without taxes)** if you purchase for **5 persons or more**.

A preferential price can be applied if you want to follow this training in your company's offices depending on the attendees number. Please, feel free to ask us at trainings@sensiolabs.com for more details.

Commercial contact: Nicolas BLIN – +33 1 40 998 109

Operational contact: Hugo HAMON – +33 1 40 998 211

PEDAGOGIC MEANS

The training takes place with a Symfony expert; who helps attendees to learn theoretical concepts and practice exercises.

All attendees need to come with their own laptop equipped with a functional LAMP stack described below:

- An Apache 2 web server
- A MySQL5 server
- PHP 5.2.4 (or later) with PDO extension
- An IDE (Eclipse, Netbeans...) or text editor like Notepad++ or TextMate

Attendees benefit from an Internet connection (wifi or Internet) to have access to the free documentation online.

Lots of exercises are realized along the training. Attendees will develop two applications. The first one is a refactoring exercise which aims to analyze and migrate a pure PHP4 application to a Symfony based application. The second exercise to develop is a small store including products list, shopping cart, contact form and backoffice administration. The application will also be validated thanks to several unit and functional tests suites.

THE TRAINER

The trainer is a Sensio Labs consultant, expert with the Symfony Framework. He has a huge professional experience with big companies project.

HOW DOES THE TRAINING TAKE PLACE?

The training takes place during three days in a dedicated meeting room. Depending on the room size, we can accept from 8 to 12 attendees. The meeting room is equipped with an Internet connection (wifi or Ethernet), a screen projector and a paperboard (or white board).

A small breakfast and lunches are included in the training every day. Everything is delivered onsite to ease attendees comfort.

DETAILED PROGRAM

FROM A FLAT APPLICATION TO AN MVC BASED APPLICATION

- ✓ Understanding the layers architecture
- ✓ Understanding the MVC pattern theory
- ✓ Moving the business logic into separated classes
- ✓ Using the alternative PHP syntax to generate templates
- ✓ Splitting the projet in several levels : application, modules and actions
- ✓ Browsing into the symfony project's files
- ✓ Using a decorator to share the repeated view's elements
- ✓ Creating links thanks to helpers
- ✓ Managing a navigation bar

TEMPLATING WITH SYMFONY

- ✓ Reading request's parameters and passing them to the view
- ✓ Exploiting helpers to generate reusable templates
- ✓ Building modular templates with partials and slots
- ✓ Managing the webdesign appreance thanks to the layout
- ✓ Exploiting CSS
- ✓ Building a reusable component to share with several modules
- ✓ Exploring symfony's available helpers (text, number, date)
- ✓ Using the view.yml file and response's methods to modify the view
- ✓ Templating best practices

SYMFONY DEVELOPMENT ENVIRONMENT

- ✓ Using the symfony's command line tools
- ✓ Using several developpment environments
- ✓ Understanding how a request is performed thanks to the webdebug toolbar
- ✓ Understanding the cascading configuration principle
- ✓ Learning the YAML syntax
- ✓ Understanding the configuration's caching and the symfony's cache:clear task
- ✓ Exploring configuration files (settings.yml, app.yml, projectConfiguration.class.php)
- ✓ Debugging a symfony application (traces, log_message)
- ✓ Installing plugins: pros and cons

THE ROUTING SYSTEM

- ✓ Understanding what is an URL
- ✓ Distinguishing the difference between a GET and a POST request
- ✓ Using Apache and its URL rewriting module for incoming requests
- ✓ Understanding the bidirectionnal routing
- ✓ Writing a routing.yml file (order, tokens, default values, requirements...)
- ✓ Distinguishing a redirect and a forward statement
- ✓ Adding a custom extension to a page
- ✓ Using the name routing rules
- ✓ Using the routing to generate a navigation tree

OBJECT RELATIONAL MAPPING ABSTRACTION

- ✓ Describing the database schema with a YAML syntax for Doctrine
- ✓ Configuring symfony to use Doctrine
- ✓ Connecting an application to a database (databases.yml)
- ✓ Manipulating business objects
- ✓ Generating a model object thanks to Doctrine
- ✓ Abstracting SQL queries in order to switch to an another database
- ✓ Writing DQL (Doctrine Query Language) queries
- ✓ Using the Doctrine generated classes (accessors, mutators, magic methods)
- ✓ Using DQL to manage objects' relations
- ✓ Refactoring the action code into the model

FRAMEWORK INSTALLATION

- ✓ Installing symfony from a sandbox, PEAR or SVN
- ✓ Understanding the symfony's SVN repository organization
- ✓ Bootstrapping a project and module
- ✓ Bootstrapping a version control management
- ✓ Upgrading symfony

ADMIN GENERATOR

- ✓ Generating and bootstrapping a module
- ✓ Understanding differences between a scaffolding and an administration
- ✓ Bootstrapping an administration interface
- ✓ Exploring the generated code
- ✓ Using the generator.yml file to edit the generated code
- ✓ Editing the list view (displayed fields, pagination, filters, tooltips, actions, virtual columns)
- ✓ Editing the edit view (admin tags, actions)

AUTHENTICATION AND ACL

- ✓ Managing alert and error messages (flashes)
- ✓ Storing data in the user's session and access to the values
- ✓ Managing users authentication
- ✓ Disconnecting a user
- ✓ Testing the credentials
- ✓ Setup the ACLs thanks to the credentials system

INTERNATIONALIZATION AND LOCALIZATION

- ✓ Identifying texts to translate (__(()))
- ✓ Translating the end user interface with an XLIFF dictionary
- ✓ Changing the user culture
- ✓ Formatting dates, numbers and currencies
- ✓ Storing information related to the culture

AUTOMATED TESTS

- ✓ Validating a piece of code thanks to a unit test
- ✓ Using Lime and the symfony command line to launch tests
- ✓ Feeding a test database with fixtures
- ✓ Validating a feature with a functional test
- ✓ Using the sfTestBrowser object to simulate a navigation between pages
- ✓ Check if an un element exists in the page thanks to the DOM CSS selectors
- ✓ Automating tests